

La reconnaissance automatique des empreintes digitales

Hasnaoui Nassim Aboubakr

Résumé:

La reconnaissance d'empreintes digitales est une technique biométrique mature pour toute application d'authentification ou de vérification d'individus. Dans ce projet de fin d'études, nous décrivons la conception et le développement d'un système automatique d'authentification d'identité par empreintes digitales qui consiste à implémenter le meilleur algorithme permettant de faire la comparaison entre plusieurs empreintes.

1. Introduction

De nos jours, l'authentification devient un des points essentiels au niveau de la sécurité des contrôles d'accès dans les sociétés ou systèmes informatiques. La reconnaissance biométrique est utilisée dans bon nombre d'applications telles que la protection de l'accès à un ordinateur, un téléphone portable, une clé USB, un établissement, des cartes bancaires...

Concernant l'empreinte digitale, c'est En 1888, l'anthropologue anglais Francis Galton introduit la notion de minuties⁴ pour réaliser la comparaison d'empreintes digitales (fingerprints matching en anglais). Quatre ans plus tard, Galton publie son ouvrage (voir [3]) où il propose une classification rigoureuse des empreintes digitales et démontre qu'il y a seulement une chance sur 64 milliards que deux individus aient une même empreinte [4]. Il existe 13 types de minuties qui nous permettent de classer les empreintes digitales.

Le principe de la reconnaissance des empreintes digitales consiste à comparer une empreinte fournie au système, à une ou plusieurs autres empreintes (les minuties) dont le système dispose préalablement dans sa base de données biométrique.

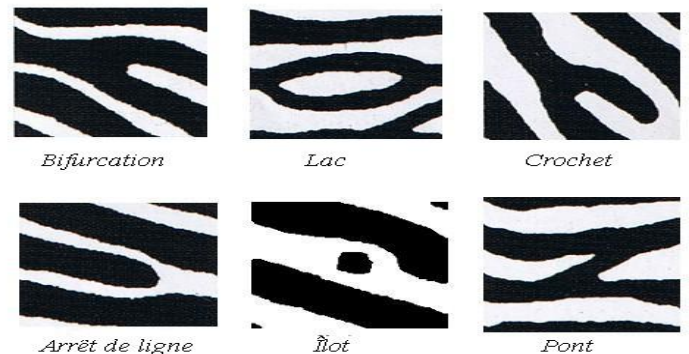


FIGURE 1: LES 6 PRINCIPAUX TYPES DES MINUTIES[5]

1.1. L'algorithme de la reconnaissance d'empreintes digitales :

Le but global est donc d'avoir un système qui fait la différence entre une image en entrée et plusieurs images situées dans une base de données. Pour cela, il faut utiliser une approche rapide et précise, c'est la raison pour laquelle on va éliminer l'approche par comparaison des images pixel par pixel parce qu'elle est assez lente.

La comparaison entre les empreintes est basée sur la recherche de la différence entre les minuties d'image d'entrée et les autres dans la base de données

La méthode généralement utilisée pour détecter les minuties consiste à mettre l'image de l'empreinte en noir et blanc, c'est la binarisation de l'image, et à donner une même taille aux lignes de l'empreinte c'est la squelettisation. Une fois que l'on dispose de l'image binaire squelettisée, les minuties (singularités) sont mieux visibles, on procède alors à leur détection.

a) Prétraitement des images d'empreinte:

Basé sur la nature des bases de données proposées par [8] qui contient les différentes empreintes on observe que tous les images nécessitent un traitement.

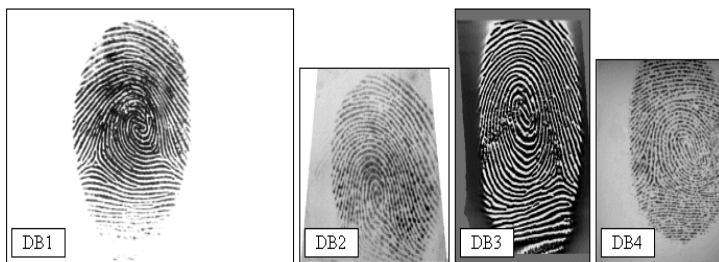


FIGURE 2: ECHANTILLANT DES BASE DE DONNES PROPOSE PAR [8]

Le premier objectif est de binariser l'image d'empreinte qui consiste à transformer une image à plusieurs niveaux en une image en noir et blanc (deux niveaux seulement).

La binarisation d'empreintes digitales est une technique pour produire une image de type 1-bit, avec 0 comme crêtes qui sont teintées de noir et de 1 les vallées qui sont teintées de blanc [9] (voir Figure 3).

Pour arriver à une image binarisée correctement il faut bien choisir une méthode de binarisation qui nous donne la forme d'empreinte sans malformation, dans ce travail on a testé plusieurs algorithmes de binarisation.



FIGURE 3: EXEMPLE D'OPERATION DE LA BINARISATION

La 2ème étape consiste à la squelettisation, l'objectif est ici de diminuer l'information redondante contenue dans une image, donc la quantité de données à analyser. La méthode est l'isolement des lignes principales de l'image avec des amincissements successifs jusqu'à ce que l'image résultante ne contienne que des lignes d'épaisseur 1 pixel.

Un algorithme d'amincissement (ou shrinking algorithm) consiste en la suppression jusqu'à stabilité de points simples, le résultat obtenu s'appelle un noyau homotypique. Si la suppression est réalisée de façon séquentielle alors la topologie est préservée ; cela par la définition même d'un point simple. Si le processus est modifié de façon à ce que certains points simples soient préservés durant le processus de suppression, il est alors possible de conserver des caractéristiques géométriques. Un tel processus s'appelle algorithme de squelettisation (ou thinning algorithm), et le résultat est appelé squelette. Les points à préserver sont appelés points terminaux ou points extrémités.[11]

Répéter

Pour tout point de l'image déterminé selon un balayage séquentiel **faire**

Si le point est simple **alors** il est supprimé

(Sinon examiner le point suivant, déterminé par le balayage)

Jusqu'à ce qu'il n'y ait plus de suppression durant un balayage complet de l'image.



FIGURE 4: EXEMPLE D'OPERATION DE LA SQUELETTISATION

b) Extraction des minuties

Après avoir obtenu l'image traitée, on doit trouver dans cette dernière les minuties les plus intéressantes de l'image.

La signature retenue pour caractériser l'empreinte est basée sur un ensemble suffisant et fiable de minuties. On entend par suffisant, le nombre minimum de minuties nécessaires pour pouvoir établir des comparaisons fiables entre empreintes. Ce minimum se situe à 12 minuties vis-à-vis de la loi, voire moins pour beaucoup d'entre eux (jusqu'à 8 minimum). Le nombre 12 provient de la règle des 12 points selon laquelle il est statistiquement impossible de trouver 2 individus présentant les mêmes 12 points caractéristiques, même en considérant une population de plusieurs dizaines de millions de personnes.

On entend par fiable, les minuties qui ne sont pas influencées par des défauts lors de l'acquisition de l'image ou par l'altération temporaire de l'empreinte digitale (blessure, érosion, etc.). Avec un petit nombre de minuties (15 ou 20) correctement localisées, il est possible d'identifier une empreinte parmi plusieurs millions d'exemplaires.

Généralement, chaque minutie occupe un espace de 16 octets sans compactage ni compression. Ceci explique la taille de chaque fichier signature, 240 octets pour 15 minuties et 1600 octets pour 100 minuties. Les bifurcations et les terminaisons sont les deux types de minuties les plus utilisés car ils sont facilement détectables, mais surtout parce qu'ils sont très aisément représentables par le modèle de coordonnées, où chaque minutie est représentée par les coordonnées (x, y).

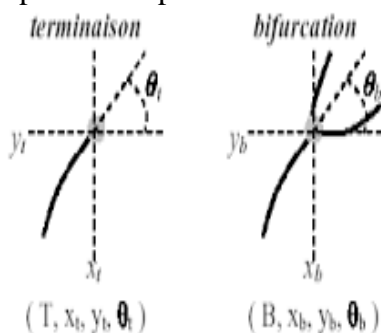


FIGURE 5: EXEMPLE DES DEUX MINUTIES LES PLUS UTILISE TER ET BIF

Les minuties sont généralement extraites à partir du squelette de l'image, il existe une approche permettant de détecter les minuties ainsi que leurs types (Terminaison, Bifurcation) en calculant l'indicateur Crossing-Number (CN) en fonction du voisinage de chaque pixel. Selon la valeur de CN le type du point est déterminé : Continuité ou Discontinuité (minutie) .[9]

Cette méthode ne retient que l'emplacement des minuties les plus pertinentes.

En analysant le squelette binaire de l'image de l'empreinte, on remarque que les pixels correspondant aux minuties possèdent un **crossing-number** différent de 2. Le **crossing-number** d'un pixel p se calcule par la formule suivante :

$$CN = 0.5 * \sum_{i=1}^8 |P_i - P_{i+1}|$$

$P_9 = P_1$, P_i est la valeur des pixels dans le voisinage 3×3 de P.

p_0, p_1, \dots, p_7 sont les 8 pixels au voisinage de p. En effet le coefficient CN présente des caractéristiques qui permettent d'identifier la nature d'une minutie en fonction du résultat obtenu lors du calcul de CN.

CN	NATURE DE LA MINUTIE EN P
0	Erreur => Point isolé
1	Terminaison
2	Erreur => Point \in Sillon
3	Divergence ou bifurcation
4	Erreur => Minutie à 4 branches

TABLE 1: IDENTIFICATION DE LA MINUTIES APARTIR DU CALCUL DE CN

2. Etat de l'art:

1.2. Les technique de reconnaissance:

i. EFinger:

Cet algorithme a été programmé par [14], présente les mêmes phases de prétraitement des images que nous déjà expliqué dans la partie précédant.

Dans cet algorithme on s'intéresse à la partie comparaison.

Efinger dispose 3 méthodes permettant d'effectuer des comparaisons d'image d'empreintes digitales :

- MIN DISTANCE
- IMAGE MAPPING
- QUAD TREE

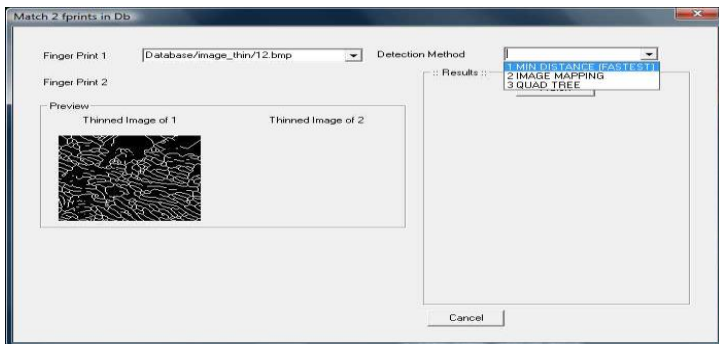


FIGURE 6: L'INTERFACE DE EFINGER DANS LA PARTIE COMPARAISON

A l'aide de l'interface proposé par [14] Figure6, ils ont choisir une empreinte et comparent à toute la base de données d'empreintes utilisons les trois méthodes de comparaison proposé.

D'après [14], la méthode **MIN DISTANCE** permet de calculer un pourcentage en comparant les valeurs similaires de deux tableaux de minuties. En effet, comme dit précédemment chaque empreinte dispose d'un fichier.TXT contenant les coordonnées des minuties de ces empreintes.

D'après [14], la méthode **MIN DISTANCE** prend en argument deux tableaux mindata1 et mindata2 à deux dimensions contenant les coordonnées x et y des minuties. Ces tableaux sont construits à partir des fichiers .TXT.

La méthode Min Distance effectue une comparaison des distances entre les minuties de l'empreinte à comparer et celle des minuties des empreintes de la base. Un score est donné pour être comparé à un seuil préétabli.

L'algorithme de comparaison **IMAGE MAPPING**, effectue quant à lui une rotation de l'image squelettisée et la compare avec celles de la base de données.

La méthode **QUAD TREE** segmente l'image en 4 parties et compare ces parties à celles des empreintes de la base de données.

D'après [14], le temps du traitement de chaque empreinte varie autour de 5 à 6 secondes.

	Min distance	Image Mapping	Quad Tree
Temps de comparaison avec 10 empreintes	Entre 0,7s et 2,60s	Entre 1,45 min et 1,53 min	Entre 0,65s et 1,10s
Temps de comparaison avec 40 empreintes	Varie entre 0,9s et 7,20 s Moyenne est autour de 3s	Varie entre 6,38 min et 6,53 Moyenne est autour de 6,44 min	Varie entre 6,38 min et 6,53 Moyenne est autour de 1,8s

TABLE 2: RESULTATS DES 3 METHODE DE COMPARAISON EFINGER

Remarque:

Comparaison avec une base de 10 empreintes :

D'après le tableau 2, la méthode « min distance » sort des résultats aléatoires pour chaque comparaison d'empreintes.

Pour ce qui concerne la méthode Image Mapping, il faut savoir qu'ils ont trouvent une tranche de temps équivalent pour les dix empreintes différentes.

Après avoir le tableau 2, la méthode Quad Tree, nous pouvons voir que celle-ci est la plus rapide au niveau comparaison des empreintes, par contre elle est moins performante en terme d'efficacité.

Comparaison avec une base de 40 empreintes :

Méthode « Quad Tree » est la plus rapide, par contre elle est moins performante en terme d'efficacité.

- L'augmentation de la base d'empreinte a une grande influence sur le temps de comparaison sur l'« Image Mapping ».

ii. Apprentissage artificiel

Un ensemble d'ingénieurs [15] Ont utilisé l'approche par réseaux de neurones pour la reconnaissance automatique des empreintes digitales.

Les étapes du prétraitement sont pratiquement similaires aux autres algorithmes, la seule différence est dans la phase de comparaison.

Les minuties de l'empreinte digitale sont extraites à partir de son squelette en calculant la «Connectivité » CN qu'on a expliqué précédemment, le nombre des minuties extraites dans cet algorithme sont 12 .

La base de données utilisée dans ce projet contient 100 images, 50 pour l'apprentissage et 50 pour le test.

Dans la partie apprentissage l'un des objectifs c'est l'apprentissage du réseau En utilisant l'algorithme de back-propagation.

back-propagation est la méthode la plus utilisé pour entrainer le réseau de neurone

Cette méthode fonctionne comme suit:

Le modèle d'entrée sur laquelle le réseau doit être entraîné est présenté à la couche du réseau d'entrée et le réseau est exécuté normalement pour voir ce que la sortie produit effectivement. La sortie réelle est comparée à la sortie désirée pour ce modèle d'entrée. Les différences entre la forme réelle et désirée un modèle d'erreur [16].

Cet algorithme utilise 12 paramètres d'entrée et 6(les caractéristique ou les minuties) paramètres de sortie. Le chiffre 6 a été extrait à partir du nombre d'apprentissage $50 \approx (2^6)$.

Dans cette étude ils ont utilisent MLP (Multilayer Perceptron) pour la classification des modèles.

D'après [15], l'extraction des caractéristiques est l'une des tâches les plus importantes pour un système de reconnaissance. MLP est conçu pour détecter les caractéristiques de l'image d'empreinte digitale de la taille 188x240 pixels. La première couche du réseau comprend 12 neurones associés aux composantes du vecteur d'entrée. La couche cachée à 25 neurones et la couche de sortie a 6 neurones. La figure 15 montre la structure en trois couches MLP.

La mise en réseau sera entraînée en utilisant l'algorithme de rétro-propagation (back-propagation).

L'étape suivante consiste à saisir les images d'empreintes digitales prototype pour extraire les caractéristiques.

- Phase de test:

Le taux de reconnaissance estimé est à 100%, ce système a été considéré comme l'un de meilleure pour la reconnaissance des empreintes digitales, c'est pour cela, plusieurs algorithmes sont basés sur ce principe.

Type	N°. échantillon	Taux de reconnaissance
Apprentissage	50 images	
Test	50 images	100%
Le taux général de la reconnaissance		100%

TABLE 3:LE TAU DE RECONNAISSANCE

3. Contribution

La méthode la plus répandue consiste à extraire les minuties à partir d'un squelette de l'image. Comme la montre la Figure 7 l'image est d'abord préparée à l'étape d'extraction au moyen d'une binarisation et d'une squelettisation, ensuite un fichier signature est extrait de l'empreinte après la détection et l'extraction des minuties Cet algorithme a été implémenté en java à l'aide d'Eclipse.

Sur la question pourquoi utiliser le langage Java plutôt que le langage C (ou C++), le Java possède des API bien plus documentée que la plupart des langages de programmation et la majorité des ordinateurs possèdent la JVM pour exécuter des programmes Java

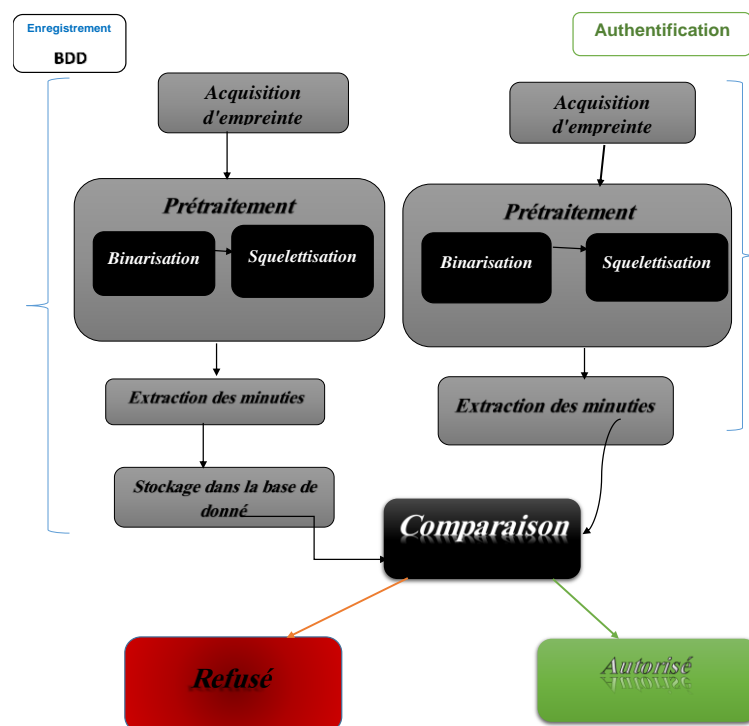


FIGURE 7:PROCESSUS SUIVIE DANS UN SYSTEM DE LA RECONNAISSANCE DES EMPREINTES

3.1. Prétraitement d'image d'empreinte

Pour permettre la squelettisation, l'image doit d'abord être binarisée, c'est-à-dire que l'image en 256 niveaux de gris dont nous disposons à ce stade est transformée en Image binaire où les pixels noirs correspondent aux stries et les pixels blancs aux vallées.

A) Binarisation d'images par la méthode d'Otsu

Une des méthodes les plus importants pour déterminer le seuil global est la sélection de seuil Otsu. Dans un seuil global, nous choisissons une valeur de seuil unique pour l'ensemble des images.

Après la phase d'implémentation avec java on remarque (Figure19) que dans les cas où la qualité d'une image d'empreinte est très faible la méthode de seuil global ne peut pas garantir des résultats interprétable est-il est nécessaire de trouvé un seuil spécial qui a un effet suffisant pour déduire une image résultant acceptable est utilisable dans la prochaine phase de la squelettisation

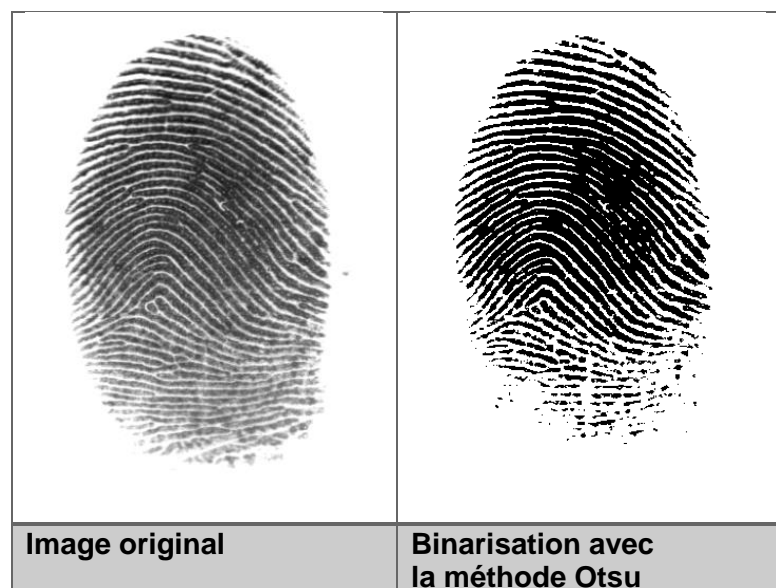


FIGURE 8:BINARISATION AVEC LA METHODE D'OTSU

✓ La solution:

Nous allons choisir des méthodes de binarisation local qu'il va appliquer le seuil dans chaque canal séparément.

Il existe autre algorithmes similaire spécialisé à résoudre le problème de la binarisation local comme:

- Bradley Local Threshold
- Bernsen Threshold.
- Maximum Entropy Threshold.

Ce sont des techniques de seuillage utilisé lorsque le fond est uniforme ou les différentes parties d'un document ont différentes origines.

A ce point nous allons choisir une des trois méthodes pour l'utiliser dans notre système.

B) Bernsen local Threshold:

On utilise des méthodes de seuillage local, Par «local», on entend ici que le seuil est calculé pour chaque pixel en fonction des caractéristiques de l'image à l'intérieur d'une fenêtre de rayon r autour d'elle.

Le procédé de seuillage local du Bernsen calcule le minimum local et le maximum pour un voisinage autour de chaque pixel.

Le procédé utilise un seuil de contraste fourni par l'utilisateur. Si le contraste local (max-min) est supérieur ou égale au seuil de contraste, le seuil est fixé à la valeur de gris moyenne locale (la moyenne du minimum et le maximum des valeurs de gris dans la fenêtre locale).

-Pour faciliter l'extraction des minuties, l'image doit être squelettisée.

Les deux méthodes de squelettisation Zhang et Shapiro qu'on a cité dans la partie précédant nous allons les expérimenter sur des images binaire, l'objectif étant d'extraire les minuties.

- D'après [19] [20], Supposant qu'on a une image 3*3 démontré comme suit :

P9	P2	P3
P8	P1	P4
P7	P6	P5

a) **l'algorithme d'amincissement de Zhang-Suen**

$A(P1)$ = nombre de pixel 1 ou 0 qui dans l'entourage de P1, dans notre cas P2, P3, P4, P5, P6, P7, P8, P9, P8.

$B(P1)$ = nombre des pixels noire ou 1 qui dans l'entourage de P1.

On ajoute la 1^{ère} condition pour sélectionner les pixels noire pour supprimer

Condition 1: $2 \leq B(P1) \leq 6$

Condition 2: $A(P1)=1$.

Condition 3: $P2. P4. P6 = 0$

Condition 4: $P4.P6.P8=0$

Cette itération est répétée jusqu'à stabilité, i.e. jusqu'à ce qu'il n'y ait plus de point simple.

b) **l'algorithme d'amincissement de Hilditch**

$B(p1)$ = nombre de voisins non nuls de p1

Et

$A(p1)$ = nombre de motifs 0,1 dans la séquence p2, p3, p4, p5, p6, p7, p8, p9, p2

Il existe deux versions pour l'algorithme de Hilditch, une en utilisant une fenêtre de 4x4 et l'autre à l'aide d'une fenêtre de 3x3.

L'algorithme de Hilditch consiste à effectuer plusieurs passes sur le modèle et à chaque passage, l'algorithme vérifie tous les pixels et décident de changer un pixel du noir au blanc si elle satisfait les quatre conditions suivantes [20]:

- $2 \leq B(p1) \leq 6$
 - $A(p1) = 1$
 - $p2.p4.p8 = 0$ ou $A(p2) \neq 1$
 - $p2.p4.p6 = 0$ ou $A(p4) \neq 1$
- Stop lorsque rien ne change (pas plus de pixels peut être retiré)

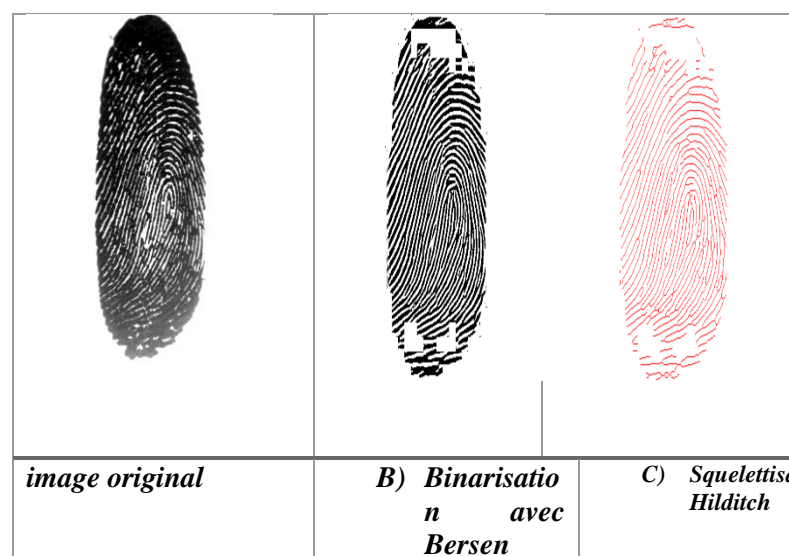


FIGURE 9: OPERATION DU PRETRAITEMENT

3.2. Extraction des minuties:

Les deux étapes de préparation à l'extraction (binarisation et squelettisation) ont grandement facilité cette phase. Nous avons utilisé la méthode de crossing number CN qu'on a déjà citée dans la partie précédant pour extraire les bifurcations et les terminaisons

$$CN = 0.5 * \sum_{i=1}^8 |p_i - p_{i-1}|,$$

avec $p_8 = p_0$ et $p_i \in \{0,1\}$

3.3. Comparaison des minuties:

Dans la conception d'un système qui fait la reconnaissance des empreintes digitale les premiers objectifs que nous pensons à atteindre c'est la précision et la rapidité, donc si on va choisir une méthode basé sur la comparaison des images pixel par pixel il peut s'avérer assez lent, donc on a utilisé une méthode qui prend les coordonnées des minuties qu'on a stocké sous format texte dans une table de bases de données, donc la partie authentification ou comparaison l'empreinte va se réduire à une simple comparaison de minuties. Il faut souligner que nous avons créé une table qui contient le **nom**, **prénom**, **id** et les **coordonnées** des minuties d'un utilisateur. Nous avons aussi saisie une base de données qui contient 82 personnes avec le SGBD SQLite.

Dans la partie d'authentification la recherche d'une empreinte parmi les empreintes de la base de données est basée sur le principe d'un moteur de recherche, les coordonnées des minuties sont stockées dans le champ *ftemplate*.

On à utiliser l'API r2xml pour faire la recherche.

Donc notre contribution au domaine de la biométrie manifesté à la création d'un system qui fait la recherche d'une empreinte à partir des cordonnées des minuties extraite.

Notre application se compose de trois grandes parties principales et d'autres fonctionnalités supplémentaires

Figure 10:

- 1- **Analyse de l'empreinte**: C'est une partie pour faire le traitement (binarisation → squelettisation) et l'extraction des minuties de façon **manuelle**, nous pouvons examiner étroitement la nature des bifurcations et terminaison
- 2- **Authentification**: Permet de faire le prétraitement et l'extraction des minuties, quand l'utilisateur met une nouvelle image d'empreinte, la fenêtre d'application affichera automatiquement l'image squelettisée avec les minuties choisie à gauche et à droite l'image original de l'empreinte et aux dessous les coordonnées des minuties dans une TextArea
- 3- **Comparer**: Cette partie permet de faire la comparaison entre deux empreintes digitales.
- 4- Cette partie présente un mini moteur de recherche qui permet de rechercher le nom d'un utilisateur dans notre base de données

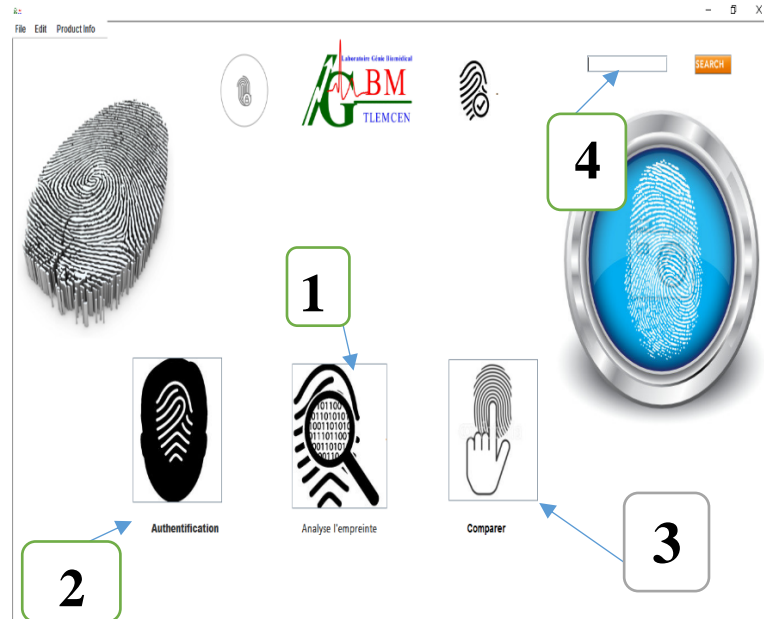


FIGURE 10: L'INTERFACE PRINCIPALE DE L'APPLICATION.

CONCLUSION GENERALE

Au cours de ce travail nous avons étudié les différents algorithmes proposés dans la littérature et les différents problèmes qui se posent durant la procédure de la reconnaissance des empreintes digitales. Nous avons développé une application pour la reconnaissance des empreintes digitales. Enfin nous n'avons traité que quelques points choisis. En particulier, nous n'avons pas abordé la reconnaissance des fausses empreintes, qui pourra faire l'objet d'un futur projet.

Référence:

- [1] Maltoni Davide, Dario Maio, Anil K. Jain, Salil Prabhakar, Handbook of fingerprint recognition, Springer, New York, 2003.
- [2] Biosentis, www.biosentis.com, consulté le 25/04/16
- [3] Francis Galton, Fingerprint, McMillan, London, 1892.
- [4] Dusenge Tony, La Reconnaissance des Empreintes Digitales, BA3-INFO Université Libre de Bruxelles, 25 mai 2009.
- [5] LA POLICE SCIENTIFIQUE, Caractéristiques d'une empreinte digitales et différenciation, <http://la-police-scientifique.e-monsite.com/> consulté le 26/02/2016.
- [6] M. Patrick ISOARDI Serrure biométrie, Reconnaissance d'empreintes digitales, Fao Frédéric-Liméry Lionel-Guiraud Ludovic.
- [7] Maltoni Davide, Dario Maio, Anil K. Jain, Salil Prabhakar, Handbook of fingerprint recognition, Springer, New York, 2003.
- [8] FVC2004 (Fingerprint Verification Competition) www.bias.csr.unibo.it/fvc2004 consulter le 25/01/2016
- [9] Christel-Loïc TISSE, Lionel MARTIN, Lionel TORRES, Michel ROBERT, Système automatique de reconnaissance d'empreintes digitales. Sécurisation de l'authentification sur carte à puce, Advanced System Technology Laboratory.
- [10] Notions de traitement et d'analyse d'image www.foad-mooc.auf.org/ consulté le 26/04/16.
- [11] Christophe LOHOU, Contribution à l'analyse topologique des images : étude d'algorithmes de squelettisation pour images 2D et 3D, selon une approche topologie digitale ou topologie discrète, Informatique Fondamentale et Applications, 20 décembre 2001.
- [12] J.R. Parker. *Algorithms for image processing and computer vision*. Wiley & Sons, Novembre 1996.
- [13] Haralick, Robert et Shapiro. *Computer and robot vision*. Vol. 1, Addison-Wesley, 1992.
- [14] Liméry Lionel, Fao Frédéric, Guiraud Ludovic, Reconnaissance d'empreintes digitales Serrure biométrie, IUP GMI D'AVIGNON, 2005.
- [15] Hamsa A. Abdullah, Fingerprint Identification System Using Neural Networks, Nahrain University, College of Engineering Journal (NUCEJ) Vol.15 No.2, 2012 pp234 - 244
- [16] A. Askarunisa, Sankaranarayanan. K, Sundaram. R and Sathick .M. Batcha, —**Finger Print Authentication Using Neural Networks**l, MASAUM Journal of Computing, Vo. 1, No. 2, 2009.
- [17] Futronic www.futronic-tech.com/product_fs80.html consulté le 02-05-2016.
- [18] N. Otsu, *A threshold selection method from grey scale histogram*, IEEE Trans. on Syst. Man and Cyber, vol 1, pp 62-66, 1979
- [19] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu and Ching Suen, Character Recognition Systems: A Guide for Students and Practitioners, AJOHN WILEY & SONS, 2007.
- [20] [Danielle Azar](#), [Pattern Recognition course](#): Hilditch's Algorithm for Skeletonization [Prof. Godfried Toussaint](#), 1997
- [21] Nicolas Galy, Etude d'un système complet de reconnaissance d'empreintes digitales pour un capteur microsystème à balayage, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, 14 avril 2005.
- [22] Nayef's Blog www.nayefreza.wordpress.com, consulté le 03/03/2016.
- [23] Les expressions régulières avec l'API Regex de Java www.cyberzoide.developpez.com/tutoriels/java/regex/, consulté le 02/05/2016.